-ll-

United States Patent and Trademark Office

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/672,975	09/26/2003	Robin Alexis Takasugi	10014268-1	3620

22879 7590 08/09/2006

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
TSAI, SHENG JEN	

ART UNIT	PAPER NUMBER
2186	

DATE MAILED: 08/09/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/672,975	TAKASUGI ET AL.	
	Examiner	Art Unit	
	Sheng-Jen Tsai	2186	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 July 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This Office Action is taken in response to Applicant's Remarks filed on July 20, 2006 regarding application 10,672,975 filed on September 26, 2003.

2. Claims 1-30 are pending for consideration.

3. ***Response to Remarks***

Applicant's remarks have been fully and carefully considered, with the Examiner's response set forth below.

(1). Applicants contend that the prior art (Hicken et al., US 6,092,149) does not teach or suggest "adding a prefetch value to a transfer length value specified in the current host command" because "the prefetch in Hicken et al. is a separate transfer that is determined and performed after the requested data is retrieved," thus Hicken et al. do not add a prefetch value to a transfer length value specified in the current host command, and then provide this sum to a data storage device, as recited in claim 1. The Examiner disagrees with this assessment for the following reasons.

First, the invention of Hicken et al. is directed toward using <u>prefetched data</u> into a <u>cache memory</u> to maximize disk drive performance based on <u>past access history</u> [abstract]. Particularly, the data prefetched in <u>previous read commands</u> is a factor in determining the type of access [figure 1G illustrates several types of access, including "full cache hit," "partial cache hit," "skip ahead cache hit," "sequential cache hit" and "no cache hit;" column 9, lines 28-67] and the amount of data to be prefetched for <u>the current read command</u>. For instance, if the data prefetched in <u>previous read commands</u> (the prefetched data is stored in the cache buffer [figure 1A, 10]) already includes

portion of the data that is requested by the current read command, then the amount of data to be prefetched for the current read command needs to be adjusted accordingly [the caching system may prescan the cache memory during prefetch to alter the prefetch amount is response to a command request (abstract); figures 8A~8E shows the details of adjusting the prefetch amount].

Second, when Hicken et al. mention that "For every read command the invention determines how much data to prefetch after the requested data is retrieved (column 10, lines 62-67)," it means determining the amount of data to be prefetched for the current read command after the requested data of the previous read command is retrieved.

This if illustrated in more details in figures 10B~10F. In figure 10B, step 725 determines if this is a "read" command, if it is a "read" command, step B2 follows. In figure 10C, step B2, step 754 determines if this is a "partial hit in prefetch and prefetch from previous command will fetch a higher LBA than the current command." If the answer is "YES," the prefetch length is adjusted to accommodate data already requested, as stated in step 756.

Third, when Hicken et al. mention that "For every read command the invention determines how much data to prefetch after the requested data is retrieved (column 10, lines 62-67)," it merely means that **determining** the amount of data to be prefetched for the current read command after the requested data of the previous read command is retrieved. It by no means suggests that the prefetch in Hicken et al. is a **separate** transfer that is performed after the requested data is retrieved, as Applicants speculate.

In fact, the requested data and prefetch data are retrieved from the disk drive [figure 1, 40] at the same time rather than separately. This is illustrated in figure 1C. In figure 1C, Step 104 determines if there is a command from the host to be processed. Step 126 computes the prefetch length, followed by Step 132, which set buffer counter (to inform the disk and to monitor how much data to be transferred from the disk for this transfer operation) and start the disk (instruct the disk to begin transfer data). Note that in the entire flowchart only one step, Step 132, involves data transferring from the disk. Thus all data transfer from the disk to the cache buffer is perfomed only once, not twice for requested data and prefetch data separately.

(2). Applicants also contend that the Examiner does not provide any citation to Hicken et al. to support the limitation of "the transfer length is the sum of the length of the requested data and the length of the prefetch data." The Examiner disagrees with this assessment for the following reason.

The Examiner cited in the previous Office Action figure 1F, which shows the requested data and the prefetched data being present in the same cache entry. Note that figure 1D further illustrates this point by showing that each cache entry (a cache entry is created each time a stream of data is transferred from the disk into the cache buffer) contains not only the LBA (Logical Block Address of the requested data), but also the PF LBA (PreFetch data Logical Block Address). In other words, each data stream from the disk drive to the cache buffer contains both requested data and prefetch data, thus the transfer length of the data stream is the sum of the length of the requested data and the length of prefetch data.

Therefore, the Examiner's position regarding the status of all claims remain the same as stated in the previous Office Action.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 1-30 are rejected under 35 U.S.C. 102(b) as being anticipated by Hicken et al. (US 6,092,149).

As to claim 1, Hicken et al. disclose **a prefetch controller** [Disk Drive Cache System using a Dynamic priority Sequential Stream of Data Segments Continuously Adapted according to Prefetched Sequential, Random and Repeating Types of Accesses (title)] **for controlling retrieval of data from a data storage device** [the MEDIA storage, figure 1A, 40] **in response to a current host command received from a host device** [Host Computer, figure 1A, 50; figure 16B shows the flowchart in which a first and a second commands are received from the host computer], **the prefetch controller comprising**:

a sequential read detector [figure 16B shows the flowchart of a sequential read detector in which a first and a second commands are received from the host computer, and a decision is made regarding if the first and the second commands constitute a sequential access based on the address of the required data (figure 16B, step 1146); <u>when a sequential read stream is recognized</u>, a different caching system path is used

to process the commands in that stream (column 11, lines 55-60)] **configured to generate a new sequential read indication for the current host command if the current host command and a previously received host command specify read operations that are non-sequential** [figure 16B shows the flowchart of a sequential read detector in which a first and a second commands are received from the host computer, and a decision is made regarding if the first and the second commands constitute a sequential access based on the address of the required data (figure 16B, step 1146); step 1150 indicates that it is non-sequential]; **and**

a transfer length generator [For every <u>read command</u> the invention determines <u>how much data to prefetch after the requested data is retrieved</u> (column 10, lines 62-67); the transferred data comprises the "<u>requested data</u>" and the "<u>prefetched data</u>" as shown in figure 1F, and the transfer length value is the sum of the length of the requested data and the length of the prefetched data] **configured to provide a first transfer length value to the data storage device if the new sequential read indication is generated for the current host command, and provide a second transfer length value to the data storage device if the new sequential read indication is not generated for the current host command** [When the adaptive prefetching is enabled and Mode Page 8, as described below, Min and Max Prefetch are 0xFFFF, the min and max prefetch are adaptively determined based on the way data is accessing in this segment. <u>If this cache access is a sequential stream</u>, both min and max prefetch are be set to the number of blocks in a segment to fill a segment's worth of data. The system then discards requested data for this command

once the data has been transferred. If this cache access is a repetitive type of access, the min is set to the blockCount of the command and the max is the number of blocks in a segment less the blockCount of the command in order to keep the requested data for possible repeated access in subsequent commands. The default values are zero for the min and blocks per segment for the max; this allows the prefetch to be interrupted as soon as a seek can be started for a new command, but fills the segment with new data if the prefetch is not interrupted (column 10, lines 62-67; column 11, lines 1-13); Figure 1G illustrates various types of accesses, including the sequential access (158), skip ahead access (166, 170 and 172), new (174 and 176), and repeated access (164 and 168); detailed description of each type of access is provided in column 9, lines 46-67 and column 10, lines 1-13. Note that only access type 158 is sequential and all the other types are non-sequential]; and **wherein the first transfer length value is determined by adding a prefetch value to a transfer length value specified in the current host command** [For every read command the invention determines how much data to prefetch after the requested data is retrieved (column 10, lines 62-67); the transferred data comprises the "requested data" and the "prefetched data" as shown in figure 1F, and the transfer length value is the sum of the length of the requested data and the length of the prefetched data].

As to claim 2, Hicken et al. teach that **the first transfer length value is larger than the second transfer length value** [the value of the first transfer length and the second transfer length depends on the length of the "requested data" and the "prefetched data" as shown in figure 1F, and the transfer length value is the sum of the

length of the requested data and the length of the prefetched data; assuming that the length of the requested data being the same for both the first and the second commands, the length of the first transfer and the second transfer would depend on the length of the prefetched data; the length of the prefetched data depends on theparameters of minimum and maximum prefetch values that are dynamically and adaptively determined as shown in figures 8E and 8B, which may result in that the first transfer length value being larger than the second transfer length value].

As to claim 3, Hicken et al. teach that **the sequential read detector comprises: operation compare logic configured to compare an operation specified in the current host command to an operation specified in the previously received host command, and generate a first indication for the current host command if the compared operations are both read operations** [command manager, figure 1B, 302; receive and decode command, figure 2A, 318; when a <u>sequential read</u> stream is recognized, a different caching system path is used to process the commands in that stream ... (column 11, lines 55-60); When <u>sequential write</u> commands are received ... (column 11, lines 65-67)].

As to claim 4, Hicken et al. teach that **the sequential read detector further comprises:**

address compare logic configured to compare a first address associated with the current host command to a second address associated with the previously received host command, and generate a second indication for the current host command if the compared addresses are indicative of sequential operations

[figure 16B shows the flowchart of a sequential read detector in which a first and a second commands are received from the host computer, and a decision is made regarding if the first and the second commands constitute a sequential access based on the address of the required data (figure 16B, step 1146); when a sequential read stream is recognized, a different caching system path is used to process the commands in that stream (column 11, lines 55-60)].

As to claim 5, Hicken et al. teach that **the sequential read detector further comprises: a sequential read indication generator configured to generate the new sequential read indication if the first and the second indications are not generated for the current host command** [figure 16B shows the flowchart of a sequential read detector in which a first and a second commands are received from the host computer, and a decision is made regarding if the first and the second commands constitute a sequential access based on the address of the required data (figure 16B, step 1146); when a sequential read stream is recognized, a different caching system path is used to process the commands in that stream (column 11, lines 55-60)].

As to claim 6, Hicken et al. teach that **the sequential read detector comprises**: **a plurality of registers** [Tables 1,2 and 4 show a plurality of registers indicating the type of access; set up registers, figure 2C, step 222] **for storing an opcode specified in the current host command** [needs at least to determine whether this is a "read" or "write" operation], **an opcode specified in the previous host command** [figure 16B shows the flowchart of a sequential read detector in which a first and a second commands are received from the host computer, and a decision is made regarding if

the first and the second commands constitute a sequential access based on the address of the required data (figure 16B, step 1146); when a sequential read stream is recognized, a different caching system path is used to process the commands in that stream (column 11, lines 55-60)], **a start address associated with the current host command** [the Logical Block Address (LBA), figure 1D, 180; "is the first logical block address of the data requested in the second command equal to the first logical block address of the data in the prefetch for the first command?", figure 16B, step 1146], **and an end address associated with the previous host command** [the Logical Block Address (LBA), figure 1D, 180; "is the first logical block address of the data requested in the second command equal to the first logical block address of the data in the prefetch for the first command?", figure 16B, step 1146].

As to claim 7, Hicken et al. teach that **the sequential read detector further comprises**:

opcode compare logic for comparing the stored opcodes [needs at least to determine whether this is a "read" or "write" operation];

address increment logic for incrementing the stored end address, thereby generating an incremented end address [address manipulation logics in place to support the determination of the various types of access shown in figure 1G including the sequential access (158), skip ahead access (166, 170 and 172), new (174 and 176), and repeated access (164 and 168), as the determination of access type is based on the beginning and ending addresses of the requested data compared to the beginning and ending addresses of the cached data as illustrated in figure 1G; figure

1D shows the LBA, OFFSET, PF LBA and END PF LBA address associated with the cache memory structure]; **and**

address compare logic for comparing the stored start address and the incremented end address [address manipulation logics in place to support the determination of the various types of access shown in figure 1G including the sequential access (158), skip ahead access (166, 170 and 172), new (174 and 176), and repeated access (164 and 168), as the determination of access type is based on the beginning and ending addresses of the requested data compared to the beginning and ending addresses of the cached data as illustrated in figure 1G; figure 1D shows the LBA, OFFSET, PF LBA and END PF LBA address associated with the cache memory structure].

As to claim 8, Hicken et al. teach that **the sequential read detector further comprises:**

a sequential read indication generator configured to generate the new sequential read indication based on outputs of the opcode compare logic and the address compare logic [figure 16B shows the flowchart of a sequential read detector in which a first and a second commands are received from the host computer, and a decision is made regarding if the first and the second commands constitute a sequential access based on the address of the required data (figure 16B, step 1146); when a sequential read stream is recognized, a different caching system path is used to process the commands in that stream (column 11, lines 55-60)].

As to claim 9, Hicken et al. teach that **the transfer length generator comprises:**

a first register for storing the prefetch value [When the adaptive prefetching is enabled and Mode Page 8, as described below, Min and Max Prefetch are 0xFFFF, the min and max prefetch are adaptively determined based on the way data is accessing in this segment (column 10, lines 62-67; column 11, lines 1-13); For every read command the invention determines how much data to prefetch after the requested data is retrieved (column 10, lines 62-67); note that the min and max prefetch represents two registers specifying the prefetch values];

a second register for storing a zero value [The default values are zero for the min and blocks per segment for the max; this allows the prefetch to be interrupted as soon as a seek can be started for a new command, but fills the segment with new data if the prefetch is not interrupted (column 11, lines 9-13)]; **and**

a multiplexer coupled to the first and the second registers, the multiplexer responsive to the new sequential read indication for selectively outputting the prefetch value or the zero value [For every read command the invention determines how much data to prefetch after the requested data is retrieved (column 10, lines 62-67); When the adaptive prefetching is enabled and Mode Page 8, as described below, Min and Max Prefetch are 0xFFFF, the min and max prefetch are adaptively determined based on the way data is accessing in this segment. If this cache access is a sequential stream, both min and max prefetch are be set to the number of blocks in a segment to fill a segment's worth of data. The system then discards requested data

for this command once the data has been transferred. If this cache access is a repetitive type of access, the min is set to the blockCount of the command and the max is the number of blocks in a segment less the blockCount of the command in order to keep the requested data for possible repeated access in subsequent commands. The default values are zero for the min and blocks per segment for the max; this allows the prefetch to be interrupted as soon as a seek can be started for a new command, but fills the segment with new data if the prefetch is not interrupted (column 10, lines 62-67; column 11, lines 1-13); Figure 1G illustrates various types of accesses, including the sequential access (158), skip ahead access (166, 170 and 172), new (174 and 176), and repeated access (164 and 168); detailed description of each type of access is provided in column 9, lines 46-67 and column 10, lines 1-13. Note that only access type 158 is sequential and all the other types are non-sequential; figure 8E shows the flowchart of determining the access type and figure 8B shows the flowchart of deciding the prefetch length depending on the access type].

As to claim 10, Hicken et al. teach that **the transfer length generator further comprises:**

a third register for storing the transfer length value specified in the current host command [the transferred data comprises the "requested data" and the "prefetched data" as shown in figure 1F, and the transfer length value is the sum of the length of the requested data and the length of the prefetched data; figure 1D shows the storage of the parameter "BLOCK COUNT"].

As to claim 11, Hicken et al. teach that **the transfer length generator further comprises**:

an adder for adding the value stored in the third register and the value output by the multiplexer the transferred data comprises the "requested data" and the "prefetched data" as shown in figure 1F, and the transfer length value is the sum of the length of the requested data and the length of the prefetched data; address manipulation logics in place to support the determination of the various types of access shown in figure 1G including the sequential access (158), skip ahead access (166, 170 and 172), new (174 and 176), and repeated access (164 and 168), as the determination of access type is based on the beginning and ending addresses of the requested data compared to the beginning and ending addresses of the cached data as illustrated in figure 1G; figure 1D shows the LBA, OFFSET, PF LBA and END PF LBA address associated with the cache memory structure; figure 1D shows the equation of "ADDRESS OF LBA = (ADDR OF SEG) + (OFFSET * SECTOR SIZE)"].

As to claim 12, refer to "As to claim 1" presented earlier in this Office Action.

As to claims 13-14, Hicken et al. teach **buffering the data** [figure 1E shows the buffer] **received from the storage device and outputting the buffered data to the host** [figure 1F shows how the requested data and the prefetch data is arranged in the buffer; figure 1G shows how data may be buffered depending on the access type].

As to claim 15, refer to "As to claim 3" and "As to claim 4."

As to claim 16, refer to "As to claim 9" and "As to claim 11."

As to claim 17, refer to "As to claim 1" presented earlier in this Office Action.

As to claim 18, refer to "As to claim 15."

As to claim 19, refer to "As to claim 9" and "As to claim 11."

As to claim 20, refer to "As to claim 1," "As to claim 12," and "As to claim 17."

As to claim 21, refer to "As to claim 2."

As to claim 22, refer to "As to claim 3."

As to claim 23, refer to "As to claim 4."

As to claim 24, refer to "As to claim 5."

As to claim 25, Hicken et al. teach that **the computer-readable medium of claim 20, wherein the method further comprises**:

storing an opcode specified in the current host command, an opcode specified in the previous host command, a start address associated with the current host command, and an end address associated with the previous host command [figure 16B shows the flowchart of a sequential read detector in which a <u>first</u> and a <u>second</u> commands are received from the host computer, and a decision is made regarding if the first and the second commands constitute a sequential access based on <u>the address of the required data</u> (figure 16B, step 1146); when a <u>sequential read</u> stream is recognized, a different caching system path is used to process the commands in that stream (column 11, lines 55-60)].

As to claim 26, refer to "As to claim 9" and "As to claim 11."

As to claim 27, refer to "As to claim 9" and "As to claim 11."

As to claim 28, refer to "As to claim 9."

As to claim 29, refer to "As to claim 10."

As to claim 30, refer to "As to claim 11."

6. ***Related Prior Art***

The following list of prior art is considered to be pertinent to applicant's invention, but not relied upon for claim analysis conducted above.

- Greiner et al., (US 6,216,208), "Prefetch Queue Responsive to Read Request Sequences."
- Kanai et al., (US 6,341,335), "Information Processing System for Read Ahead Buffer memory Equipped with Register and Memory Controller."
- Kaneko et al., (US 6,427,184), "Disk Drive with Prefetch and Writeback Algorithm for Sequential and Nearly Sequential Input/Output Streams."
- Bates, Jr. et al., (US 6,253,289), "Maximizing Sequential Read Streams While Minimizing the Impact of Cache and Other Applications."
- Desai et al., (US 6,789,171), "Computer System Implementing a Multi-threaded Stride Prediction Read Ahead Algorithm."
- Yu et al., (US 6,606,717), "Cache Control method and System for Mixed Streaming and Non-Streaming data."
- Henry et al., (US 6,917,990), "Method and Structure for Read Prefetch in a Storage Complex Architecture."

Conclusion

7. Claims 1-30 are rejected as explained above.

8. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sheng-Jen Tsai whose telephone number is 571-272-4244. The examiner can normally be reached on 8:30 - 5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matthew Kim can be reached on 571-272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Sheng-Jen Tsai
Examiner

PIERRE BATAILLE
PRIMARY EXAMINER
8/2/06